Applying dynamic relaxation techniques to form-find and manufacture curve-crease folded panels

Abstract

The research incorporated in the paper stems from the design and fabrication of a self-supporting, multi-panel installation for the Venice Biennale 2012 and operates against the backdrop of the exciting potentials that the field of curved-crease folding offers in the development of curved surfaces that can be manufactured from sheet material. The two main challenges were developing an intuitive design strategy and production of information adhering to manufacturing constraints. The essential contribution of the paper is a proposed interactive form-finding method for curve-crease geometries that could negotiate the multiple objectives of ease of use in exploratory design, and manufacturing constraints of their architectural-scale assemblies.

Acknowledgements.

This research was carried out with the support and guidance of Zaha Hadid Architects, especially Zaha Hadid and Patrik Schumacher.

Image 1: The completed sculpture at the Venice Biennale 2012

Keywords: Curved Crease Folding, Interactive Simulation, Design Workflows & Digital Tools, Dynamic relaxation, Multi-Objective Solvers & Feedback

1. INTRODUCTION

The research stems from the design and fabrication of a self-supporting, multi-panel installation for the Venice Biennale 2012 and operates against the backdrop of the exciting potentials that the field of curved-crease folding offers in the development of curved surfaces that can be manufactured from sheet material. The two main challenges were developing an intuitive design strategy and production of information adhering to manufacturing constraints.

The essential contribution of the paper is a proposed form-finding method for curve-crease geometries that could negotiate the multiple objectives of ease of use in exploratory design, and manufacturing constraints of their architectural-scale assemblies.

There are several seminal design and art precedents within this field - Richard Sweeney (Sweeney, R., 2013), David Huffman (Huffman, D., 1996), Erik Demaine (Demaine, E., 2010) etc. Most of the precedents projects and available literature on design methods highlight the difficulty in developing an intuitive, exploratory digital-design method to generate feasible 3D geometries. Our initial survey of methods included both the simple and common method – the method of reflection (Mitani and

Igarashi, 2011) – and the involved Planar-Quad-meshes and optimization-based method (Kilian et al., 2008). Most methods, including the two above, presented difficulties towards incorporation within an intuitive, edit-and-observe method of design; The first one proving difficult to explore variety of generalized solutions free of prior assumptions and the second one being elaborate involving scanning of physical paper models, proprietary optimization algorithms etc . For an extensive overview on the precedents, and computational methods related to curved crease folding, we refer the reader to a survey (Demaine et al., 2011). Further, we were particularly interested in the recent developments of physically-based, interactive tools that operate on user-specified coarse linear piecewise complexes that are iteratively subdivided and perturbed to produce feasible solutions via energy minimization methods (Solomon et al., 2012). This is in alignment with established benefits of subdivision surface based modeling paradigm in architectural *form-finding* (Shepherd and Richens., 2009 , Bhooshan and El Sayed., 2011) and the application of *dynamic relaxation* (Barnes, 1999) techniques on subdivision surfaces to design and fabricate minimal mean curvature surfaces - so called *minimal surfaces* (Bhooshan and El Sayed, 2012) (Figure 1).

The method proposed in this paper follows from these observations, and an explicit intention to perturb input 3D geometries to *find* feasible geometry as opposed to *finding* the folded state of a 2D input mesh. It may be noted that the optimization-based method proposed by (Kilian et al., 2008) does in fact solve this problem, albeit it is more difficult to implement. Our method is easier to implement and extend. However, unlike their method relies on the designer to provide an initial mesh with appropriate topology. We show simple procedural methods involving known mesh-operations that can be used to produce the initial mesh and the subsequent use of dynamic relaxation (DR) techniques to iteratively perturb the surface towards minimal Gaussian curvature and local planarity (Section 2). The paper will proceed by describing key discoveries made in applying *DR* to design individual panels with a few crease folds , and the subsequent incorporation of those discoveries in the design and manufacture of self-supporting, multiple-panel configurations (Section 3).

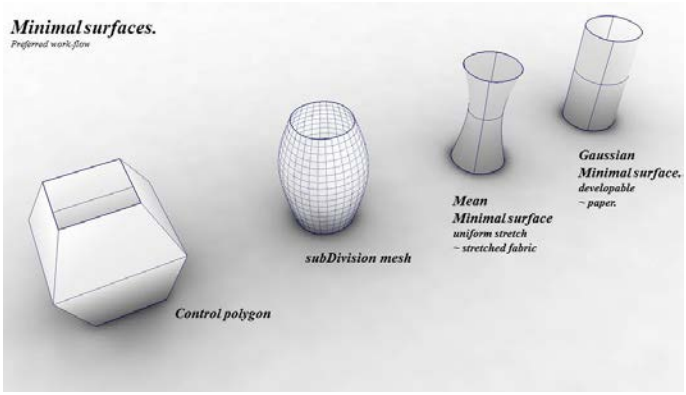


Figure 1: Showing preffered design method of starting with a coarse mesh, subdividing it and 'relaxing' it minimal conditions – either minimal mean curvature or minimal Gaussian. Also indicates indicated manufacture method.

2. COMPUTATIONAL METHOD FOR INDIVIDUAL PANELS.

2.1. Discrete representation and method overview

There are several discrete representations - *exact* and *inexact* - of curve-crease folded geometries. We chose to use a representation based on planar-quad meshes (PQ mesh) that additionally incorporate *developability* constraints (Killian et al, 2012) (Figure 2). For a comprehensive list of representations, we refer to (Solomon et al., 2012).

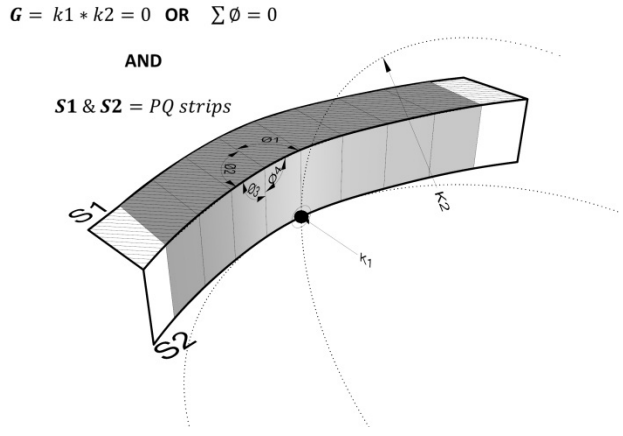


Figure 2 Showing essential requirements of a discrete representation.

Given the representation, our proposed design-friendly method essentially involves the use of various mesh operations to describe a coarse and predominantly quad-faced mesh (*low-poly*) with an appropriate topology (Figure 3) and subsequently perturbing the vertices of such a mesh towards a vanishing Gaussian curvature and the faces of the mesh towards planarity.

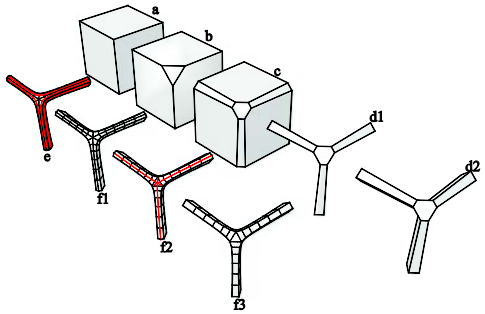


Figure 3: Shows the use of a *chamfer* and *bevel* Conway operators (Conway J., 2008) on a cube, the corresponding subdivision surface and derived topology of rulings of a tri-fold panel. The steps are as follows:
a. Closed polyhedra, preferably planar.
b. Chamfering of vertices of the closed polyhedron
c. Bevel original egdes.
d. Deleting the original faces and extrusion of border edges.
e. Conversions into a higher resolution mesh using the Catmull-Clark sub-division scheme.
f1, f2,f3. Re-topologise based on the heuristics that rulings don't intersect except at conical parts. In this case, deleting the highlighted edges & vertices

Thus, the DR-based method for the design of individual panels could be summarized (Figure 4) as below:

1. (Procedural) generation of input mesh.
2. Applying virtual forces of planarity and developability, and solving for the equilibrium positions of the input mesh subjected to such forces.
3. Computing a planar development. (2D mesh)
4. Optionally, rectify the residual 'errors' by minimizing the strain energy between the 3d and 2d meshes.

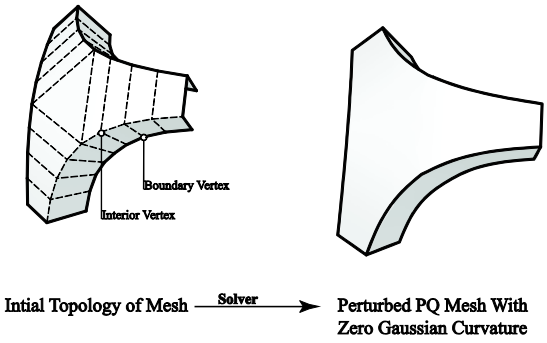


Figure 4 shows the forces and fixing scheme

2.2. Perturbation

Given the representation, and our first step of generating an input mesh of appropriate topology of rulings, the method distills to a minimization problem, stated as below:

$$\left\{ \boldsymbol{x} \mid x_i \in \boldsymbol{R^3}, argmin\, f(\boldsymbol{x}) = g(\boldsymbol{x}) + h(\boldsymbol{x}) \right.$$
$$\left. = \sum_{i=0}^{n} g_i + \sum_{j=0}^{m} \sum_{k=0}^{o} d_k \right\}$$

Where the mesh has *n* vertices and *m* faces each with *o* vertices, **x** is a *n* x 3 matrix of vertex positions, $g(\mathbf{x})$ measures angle defect at each vertex and $h(\mathbf{x})$ measures the distance of each vertex of a face to the corresponding best-fit plane.

It follows that we could iteratively perturb the vertices along the gradient of the two functions to minimize the error functional (Figure 5),

$$\nabla f(\boldsymbol{x}) = \nabla(\, g(\boldsymbol{x}) + \, h(\boldsymbol{x})\,) = \, \nabla g(\boldsymbol{x}) + \, \nabla h(\boldsymbol{x})$$

The perturbation of the vertices of the mesh follows a *dynamic relaxation* schema – treating the vertices of the mesh as lumped masses, applying virtual-forces along the respective gradients – described next - at the vertices, and subsequently updating the positions of the vertices by integrating the resulting ODE (Ordinary Differential Equation), using an appropriate numerical method.. This process is

continued until the resultant force at each vertex is zero.

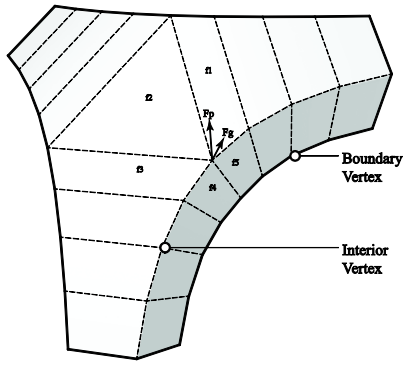


Figure 5 shows the forces and their formulations.
Interior vertex: Forces along Gradients of gaussian curvature and planarity.
Boundary vertex: Fixed or zero force.

2.3. Gradient of developability

Develop-ability can be ensured by the presence of uniform and zero Gaussian curvature throughout the mesh. A well-established discrete measure of Gaussian curvature is proportional to the sum of the angles subtended by the edges meeting at a vertex (Aleksandrov & Zalgaller, 1967). Thus the solver accumulates a force along the direction of gradient of Gaussian curvature at each vertex, with a magnitude proportional to the angle deficit (from 2π) at the vertex. We tested both an analytically computed (Desburn et al., 2002) and a numerically computed gradient, with the analytical gradient predictably converging 2-3 times faster.

$$\boldsymbol{F}_g = \boldsymbol{\nabla}\emptyset \ * \ (2\pi - \sum\emptyset)$$

where $\boldsymbol{\nabla}\emptyset$ is the gradient of gaussian curvature

2.4. Gradient of planarity

Planarity of the faces of the input mesh is ensured by accumulating forces on the vertices of each face. The direction of the force is towards the best-fit plane and its magnitude is proportional to distance of the vertex from the best-fit plane. We compute the normal of this plane as the weighted sum of the cross products of cyclical vector pairs from the node to each of its neighbours (Figure 6). The centroid of the 1-ring is considered the origin of the plane. An Eigen decomposition of the covariance matrix of the nodal positions of the neighbours may also be employed for this purpose, as noted in (Poranne et al, 20l3). We found, that the first approximation is more compatible with the developability force in achieving convergence.

The normal $\boldsymbol{N}_f and$ origin $\boldsymbol{C}_f$ of the best $-$ fit plane of each face $is\ given\ by$

$$\boldsymbol{N}_f = \sum_{i=0}^{f_e} \boldsymbol{n}_i \ , \qquad \boldsymbol{C}_f = \sum_{i=0}^{f} \boldsymbol{v}_i$$

$\boldsymbol{n}_i$ is the normal of each triangle in the face, and f is the number of vertices of the face

Then distance d of each vertex from the best-fit plane is

$$d = \left(\boldsymbol{V} - \boldsymbol{C}_f\right).\boldsymbol{N}_f$$

And the corresponding force

$$\boldsymbol{F}_{f\_v} = d * \boldsymbol{N}_f$$

Lastly, the accumulated planarity force at each vertex is

$$\boldsymbol{F}_p = \sum_{f=0}^{vf} \boldsymbol{F}_{f_v}$$

where vf is the number of faces that the vertex belongs to.

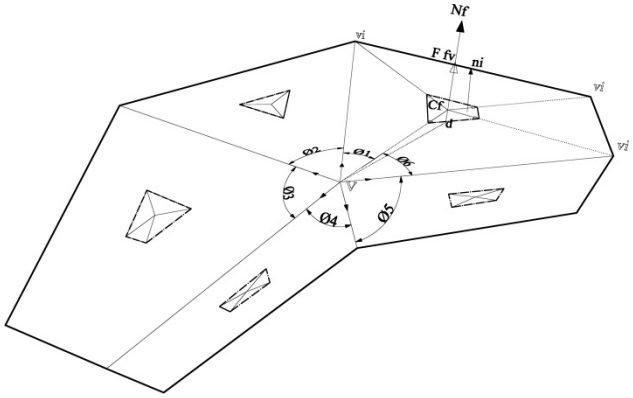


Figure 6 shows the formulation of the virtual planarity force

2.5. Boundary conditions and additional degrees of freedom

Typically, when DR is used to *form-find* minimal (mean curvature) surfaces, some or all the boundary vertices of the mesh are held fixed. However in the case of *form-finding* curve-creased surfaces, it is required to allow the boundaries to find their equilibrium positions, since physically, any folding across a crease causes the boundaries to rearrange itself to compensate for the induced stretching of the material. As such, we apply planarity forces to the boundary vertices as well. However, since the boundary vertices do not have the full set of vertices to compute the gauss curvature correctly, we do not apply any develop-ability force to those vertices.

Additionally, the boundary vertices could be considered as extra degrees of freedom that can be utilized towards finding the equilibrium solution i.e. they can be moved to affect the Gaussian curvature of the adjacent interior vertex. We thus compute a gradient of Gaussian curvature of the adjacent interior vertex, but measured at the boundary vertex(Figure 7). The analytical form of the gradient is shown below.

$$\nabla\emptyset_{bg} = \sum_{i=0}^{n} \nabla\emptyset_i = \sum_{i=0}^{n} (\boldsymbol{e}_{i1} \otimes \boldsymbol{e}_{i2}) \otimes \boldsymbol{e}_{i1}$$

Where *n* is the number of triangles common to the boundary vertex *bg* and its connected interior vertex *g*, and $\boldsymbol{e}_{i1}$ and $\boldsymbol{e}_{i2}$ are unitised vectors along the edges of each of those triangles.

Gradient of each of the angles is ortho-normal to the edge connecting the boundary vertex to the interior vertex, and the normal of the triangle. A force is then applied to the boundary vertex along this gradient direction.

$$\boldsymbol{F}_{bg} = \nabla\emptyset_{bg} * (2\pi - \sum\emptyset_g)$$

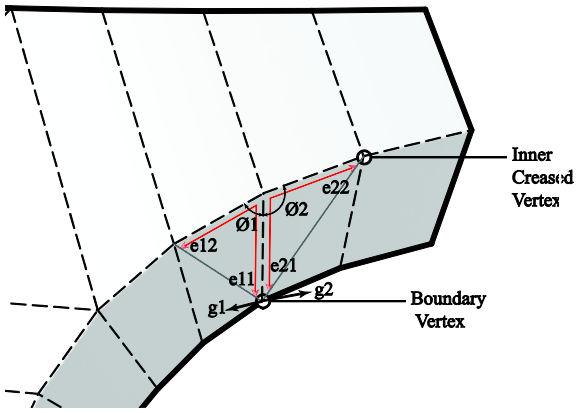


Figure 7 for boundary vertex and associated gradient.

The difference that this additional force makes to the final solution can be seen in Figure 8.

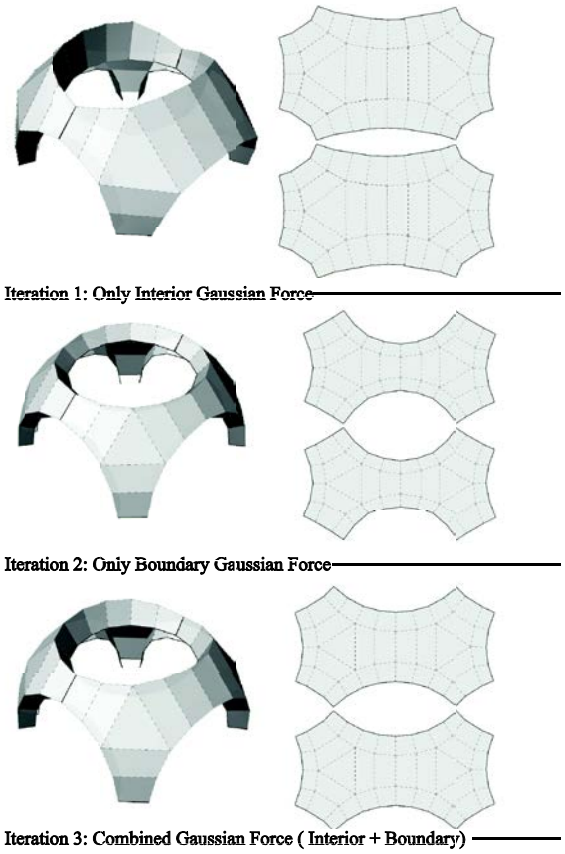


Figure 8 differences caused by various force combinations

Alternatively, we can add a row of boundary faces to the mesh, thus converting the original boundary vertices to interior ones. The extra vertices can then

be held fixed. These extra faces can be used to direct the solution of the solver, since the additional row controls half the number of angles subtended at a vertex, thereby limiting the scope of movement of the vertex in order to ensure that the total sum is 360 degrees. The effect of this user-defined addition can be seen in Figure 9.

Figure 9 shows three variants of adding extra faces – in plane, extrude normally out and normally in and the resulting equilibrium meshes

2.6. Damping

In line with the typical usage in a DR scheme, we use nominal amount of damping for the nodal velocities.

2.7. Stiffness of ODE

The *solving* for the equilibrium position of the vertices that are subjected to virtual forces of planarity and developability as described previously, is a so called 'stiff' problem i.e. it requires that the step sizes are very small. If the mesh is subjected to only one of the forces, the solver converges rapidly, and within acceptable numerical error. A possible method to overcome the stiffness is minimizing one of the functions beyond a threshold before applying the forces corresponding to the other. In practice, we have found that solving first for develop-ability and subsequently solving for planarity speeds up the search.

2.8. Planar Development and strain minimization

There is extensive literature especially in areas of computer graphics and cartography, related to this process of unfolding or computing a parametric mapping between a discrete mesh and its isomorphic and planar counterpart. We refer the reader to (Desbrun et al, 2002) for various methods used to establish such a parameterization and their limitations. We can also recommend (Sheffer et al ,2006) for a more recent survey on mesh parametrization. It is sufficient to state here that there are no perfect methods to compute a completely accurate mapping (except in the case of developable surfaces), and all current methods aim to minimize the deviations between the 3D mesh its planar counterpart – some aim to minimize angular deviations (*conformal* mapping), others preserve areas (*authalic*). It can then be stated that we required our mapping to be *isometric* i.e. both *conformal* and *authalic*.

In keeping with our DR scheme, *conformal* and *authalic* parameterizations could be achieved by applying forces to vertices along the direction of respective gradients. These gradients at each vertex are computed from the position vectors of the 1-ring neighborhood of vertices, as formulated in (Desburn et al., 2002). It may be worth noting that the methods that (Desburn et al., 2002; Pinkall and Polthier, 1993) propose, minimizes a linear energy functional but suffer from arbitrary boundary parametrization. Alternatively, the iterative methods of **A**s-**R**igid-**A**s-**P**ossible parametrization (Liu et al., 2008) and **M**ost **I**sometric **P**arametrization**S** methods proposed by (Hormann and Greiner, 2000) produce more 'natural' boundaries albeit at the expense of computational non-linearity.

However, we eventually chose to follow a spring-energy minimization method to ensure an *isometric* parameterization. This is in line with one of prominent algorithmic themes to ensure developability of meshes: minimize the strain energy between the 3d mesh and its corresponding *planar-development* (Killian et.al 2008; McCartney et al, 1999; Wang et al, 2002). Similar to (Wang et al, 2002), we achieve this by accumulating *spring* forces on vertices, proportional to difference in lengths of the corresponding edges in the 3d and 2d meshes. The differences between the three are shown in Figure 10.

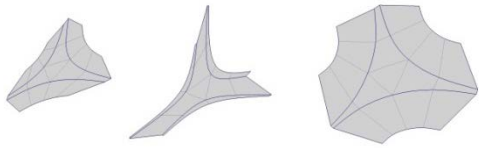


Figure 10 showing planar developments. (L to R) conformal, authalic, isomertric

For more on parameterizations with spring-like energy minimizations, we refer the reader to (Zhong & Xu, 2006). This method was particularly amenable to dissipate error that tended to accumulate in design of multiple-panel assemblies (~500 panels).This is described in greater detail in the next section.

Figure 11 Shows Various input meshes and resulting output meshes from our algorithm.

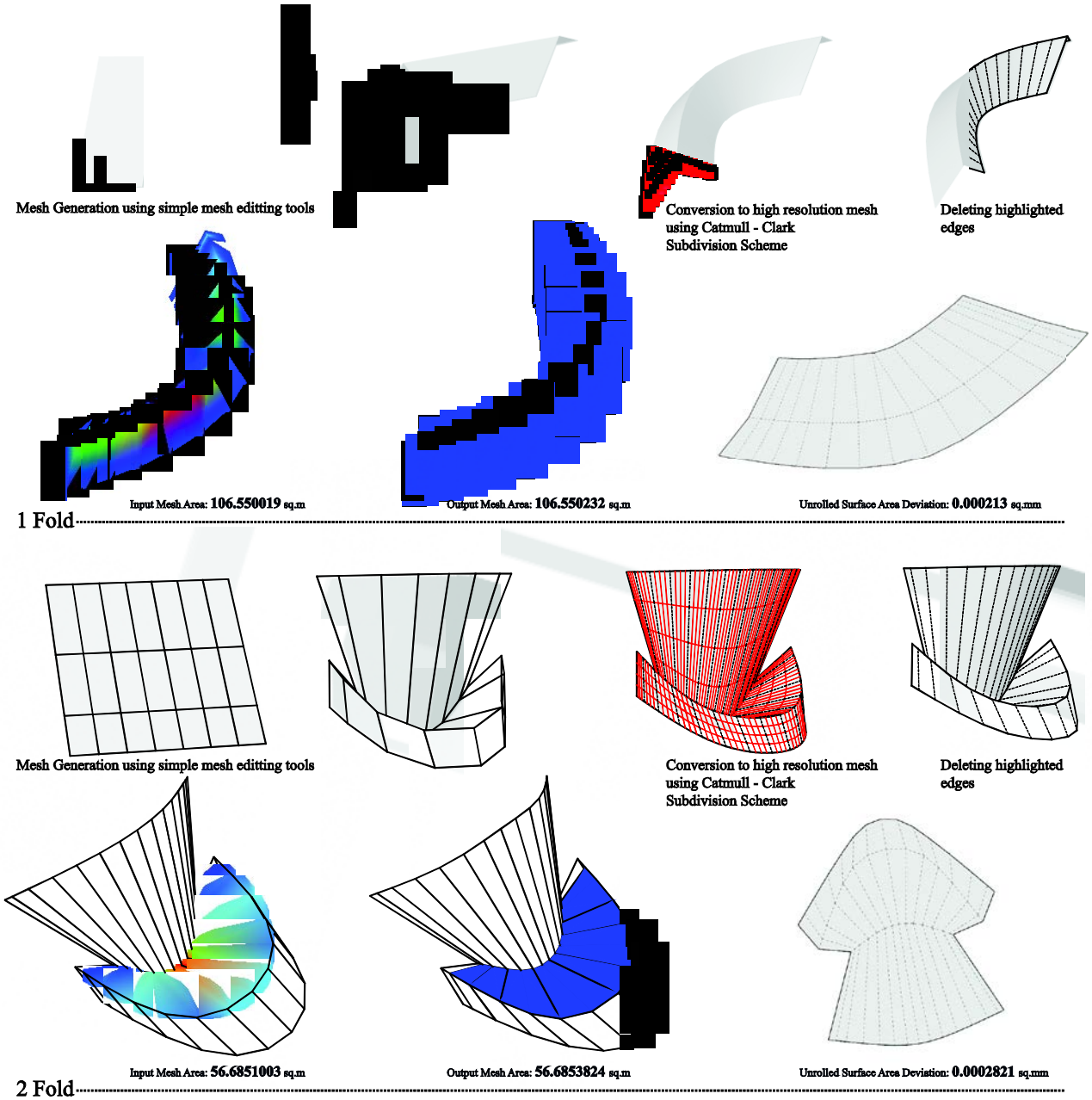
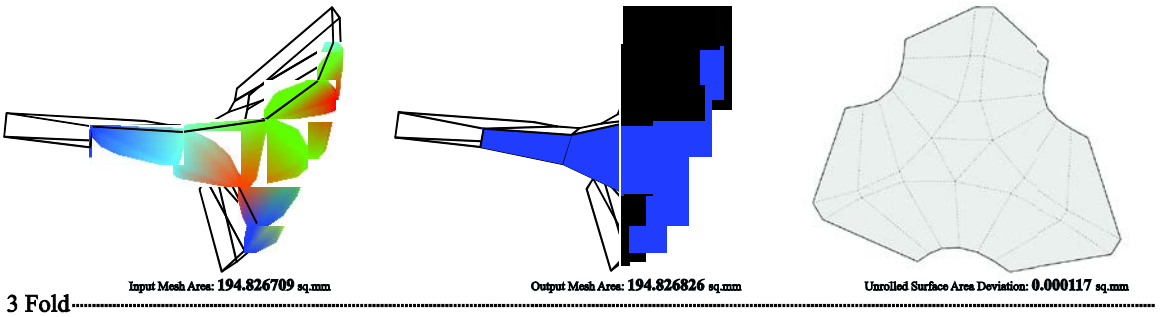
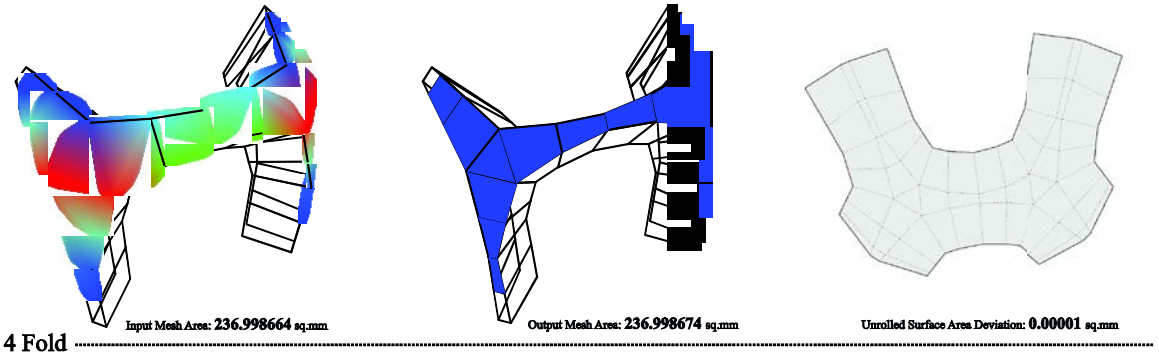


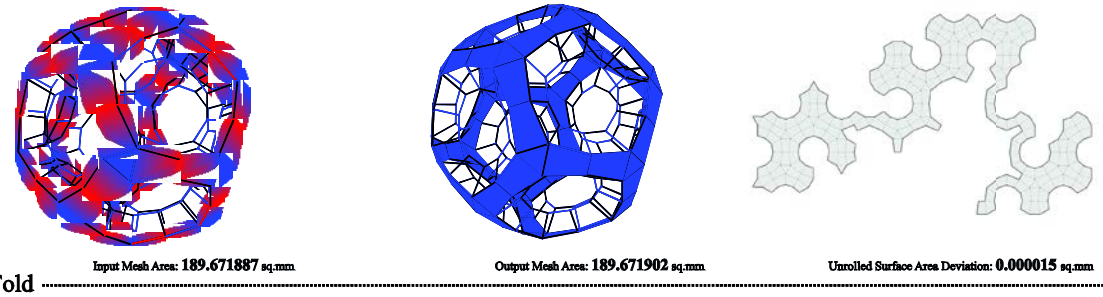
Figure 12 shows various input meshes and resulting output meshes from our algorithm



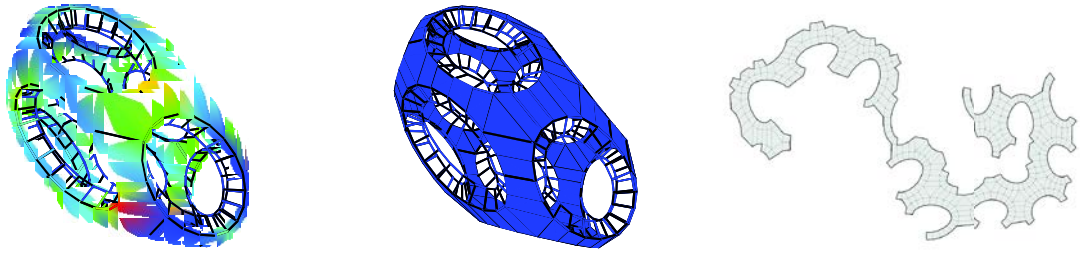
3 Fold



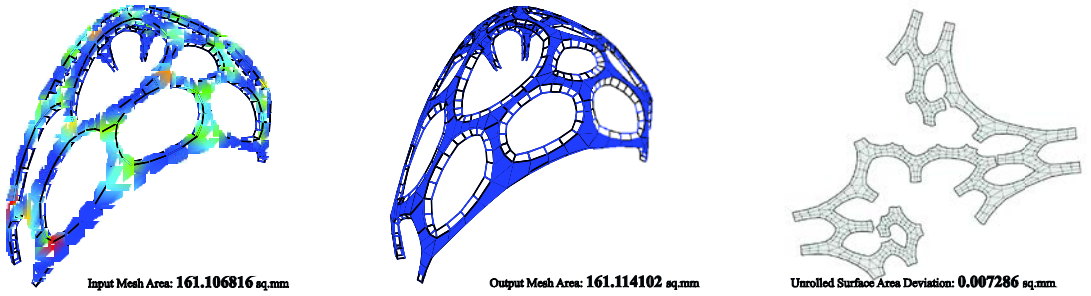
4 Fold



5 Fold



Closed Polyhedra



3 Fold - Multiple Panel

3. COMPUTATIONAL METHOD FOR MULTIPLE-PANEL CONFIGURATION.

The discoveries mentioned above were incorporated into the work-flow for the design of the multiple-panel configurations that eventually lead to the physical installation (Figure 13). The individual steps were either fully or partially-automated. However, due to constraints of production and time, the data transfer between individual steps was manual, requiring minor adjustments to correct any accumulated errors.

Figure 13 showing over-all work stages for design of multi-panel configurations

3.1. Input Mesh

The initial stage involves the interactive editing of a predominantly quad faced, low resolution mesh (*low-poly*), which captures design intent. Through this process of shaping the *low-poly*, we were able to explore and finalise essential topological conditions such as touchdown points, boundaries etc. This also allowed us to incorporate the hueristically understood structural action into this early design of the geometry. In addition to these advantages, working with a relatively low resolution mesh allows the generation of a reasonable estimate of topology; this is done by ensuring that one of the two primary directions of the mesh face flow (low-poly in Figure 14) is shaped by no more than 3 vertices thereby guaranteeing the formation of planar arcs upon subdivision. The planarity of these arcs aids in the generation of a mesh whose faces are nearly planar. Subsequently, this *low-poly* is converted into a high resolution mesh that inherits the underlying quad structure using the modified Catmull-Clark subdivision scheme Autodesk Maya uses (Stam, J., 1998). This inheritance allows the high-resolution quad grid to retain the underlying near-planarity of the faces, which reduces the need for extensive perturbation in the next steps. This high resolution mesh is then used as the input surface for the next stages in the iterative workflow therefore transferring its 'planarity' advantages to the geometries derieved from it.

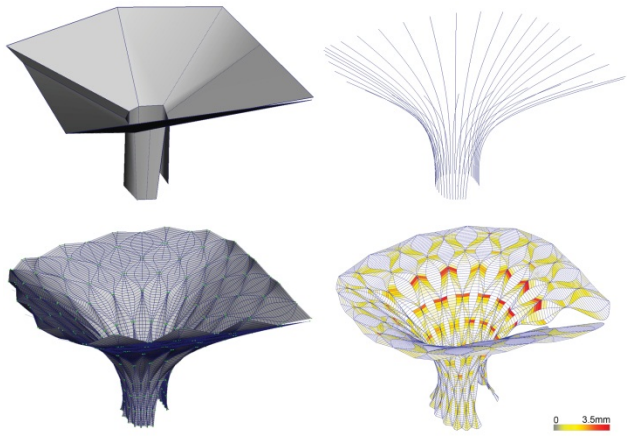


Figure 14 showing procedural generation of input mesh and out-of-plane measure of the generated mesh

Furthermore, since the design was to be materialized using thin curved-crease folded metal panels, much of the fabrication accuracy would reside in the design and generation of the panel geometries. With this in mind, a series of planar degree-3 curves were extracted from the high resolution mesh. These arcs were used to generate the panel geometries on the surface by incrementally traversing the

parametricized curve lengths of neighboring-pairs of arcs (Figure 14).

3.2. Perturbation

It is important to note that this staggered arrangement of panels, induces conical elements with their apex at the vertex between four panels. This point will be hereafter referred to as a *singularity point*. The total number of singularity points is *n/2* if *n* is the number of panels used to populate the surface (Figure 15). These singularities become fixed points in the perturbation based solver and thus allow for each panel to be petrubed relatively independently.

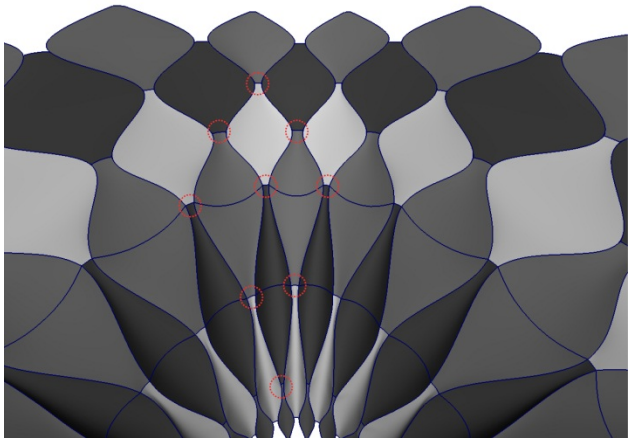


Figure 15 showing Panel singularities

3.3. Unfold and error adjustment

Each panel is subsequently unfolded into their respective flat configurations by minising the associated spring-energy, resulting in a 2D configuration of panels (Figure 16). The measure of error at this step consists of two metrics: The first is an average deviation between respective edge lengths in 3d and 2d, across the multiple panel configurations. The second is the maximum of such values.Additionally, it is relatively simple to identify the panels with the maximum errors that exacerbate the overall average of the configuration, due to the visualization of these metrics on the mesh. Subsequently, an ‘adjust and measure’ workflow is used to minimize the error metrics; this involves the manual adjustment of the local planes of the problematic panels in 3D and observing the subsequent effect on the error metrics.

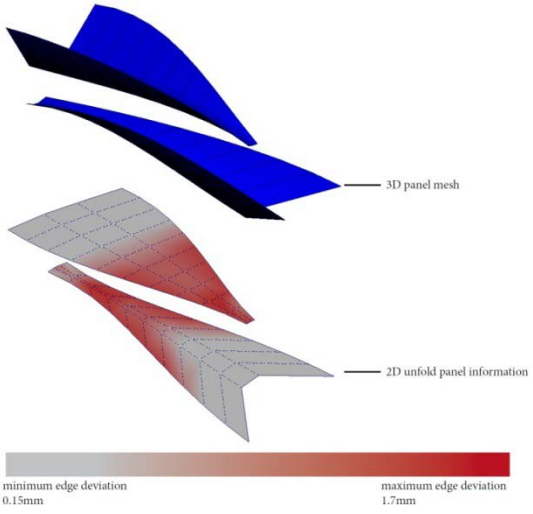


Figure 16 showing panel unfold visualizations

3.4. Edge congruency to measure unfold error

An adjacent panel border edge congruency check was used to check errors between panels in addition to error checking built into the unfold method that operates on individual panels. An automated sequence of transformations was applied to each panel so as to arrange them in adjacency ‘bands’ in order to measure the deviation between adjacent edges (Figure 17).
As an unintended but welcome consequence of this arrangement, plotting the magnitude of unfolds error across the panel edges and bands gave us a holistic overview of its relationship to surface topology. This feedback was valuable when we made adjustments to the input mesh (as described in 3.1) over several iterations to progressively minimize error.

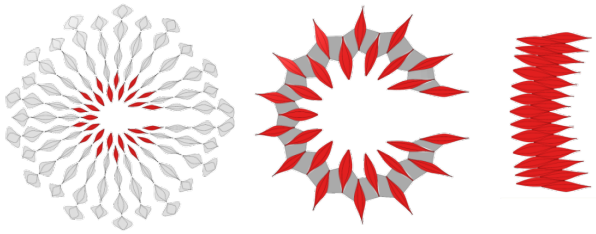


Figure 17 showing rearrangement of panels to form adjacency bands

3.5. Merging adjacent incongruent boundary edges

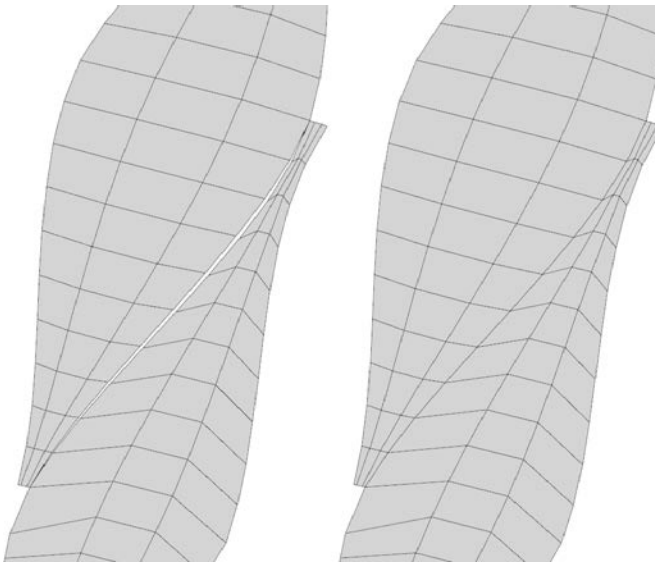


Figure 18 showing panel boundary edges before and after the merging process (above and below)

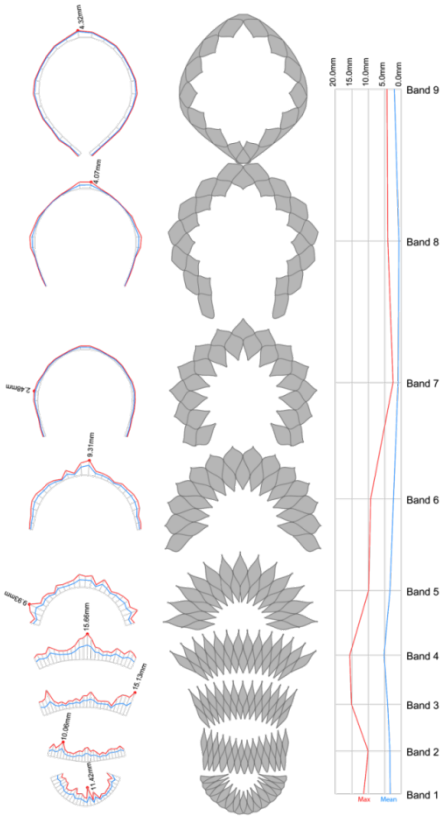


Figure 19 showing plotting unfold error per panel edge and per band provided a holistic overview of its relationship to surface topology

The data sets acquired from measuring deviation between adjacent boundary edges were per boundary edge, and to be compatible with the unfold process (as described in 3.3) they needed to be converted into per panel information. By merging the vertices of incongruent boundary edges, the gaps and overlaps between these edges were absorbed into the edge lengths of the panels themselves (Figure 18), thereby providing a new data set of target edge lengths. In certain exceptional cases with very high deviation, vertices were adjusted manually until the maximum deviations were within acceptable fabrication tolerances. Given more time, this information could be fed back into the unfold solver to automatically iterate until the manufacturing tolerances are met.

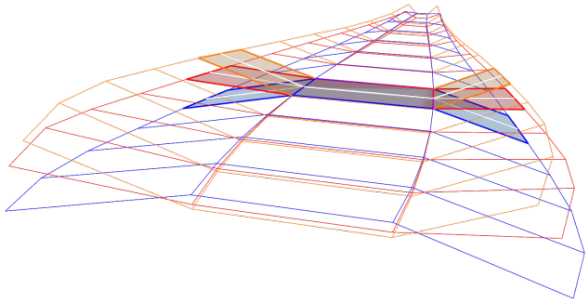
3.6. Refold

A refold method was required to produce sequential fold-state meshes for each panel in order to generate a toolpath for robotic fabrication. We opted for a method that computed folds by applying rotations constrained along edges to the mesh vertices, similar to the one proposed by Tachi and used in his Rigid Origami Simulator (Tachi, T., 2009). The algorithms non-iterative nature reduced solving time and its weighting of fold-angle helped achieve an acceptable degree of precision. Our method differed from Tachi's in that it was setup to only require fold-angles at the middle row of faces of the panel, computing the remaining vertices, edges and angles as a consequence of *actuating* the first set of folds. This ensured the highest angular accuracy at the middle row of faces that were to be used for generating the toolpaths (Figure 20). However, similar to Tachi's method, our method also required the triangulation of all quad faces as vertices with a valency of 4 did not offer enough degrees of freedom. Figure 20 showing fold Angles were only specified at the middle row of faces as the toolpaths were generated from them

4. DISCUSSION AND FURTHER WORK

4.1. Automated feedback

One of the benefits of such a multi-stage workflow is that it eases manual or algorithmic intervention as data is passed from one stage to another. In its current state, most feedback was visual and incorporated manually (figure 21). However, we think there is opportunity to automate



the feedback process to some degree as discussed briefly in section 3.5. The simplest to implement would be to use the data (shown previously in Figure 19) from step-7 and use it to drive the next iteration of step-5. A more complex but perhaps more holistic feedback loop would be to use the refolded meshes at step-8 to regenerate the next iteration of the high-resolution mesh at step-2. This was briefly attempted but the refolded meshes presented vast misalignments when places adjacent to each other due to the refold algorithm being systemically different from the rest of the perturbation-based solvers.

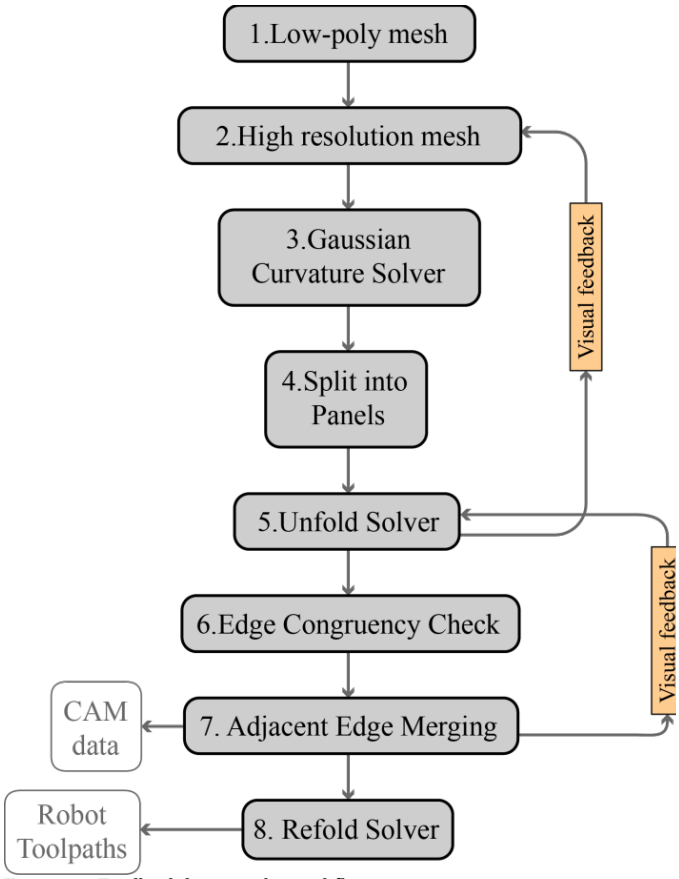


Figure 21: Feedback loops in the workflow

4.2. Pertubation and geometry

Another opportunity to improve the workflow would be to use only half the mesh geometry, due to the mirror symmetry of the form. This would have improved and addressed two issues arising from the use of perturbation based methods. The first, an improvement of the computation time of each iteration, due the reduction of input geometries. The second is that resultant geometries of the perturbation methods did not exhibit symmetrical properties, unlike their corresponding inputs. By mirroring the resultant half mesh this problem would be prevented, resulting in a symmetrical mesh that also satisfies the solvers constraints

4.3. Implicit modeling of crease-curves

It may be noted that the proposed method, does not explicitly model crease lines or fold angles across them. This is both an advantage and a limitation in the design process. The advantage is that the formulation of the solver and intuitively understanding it is simple. The method of directing it towards desired results is via interactively modifying input meshes and observing the results. This could also become a disadvantage in certain fold configuration where equilibrium solutions might be harder to find.

4.4. Mass-spring simulation

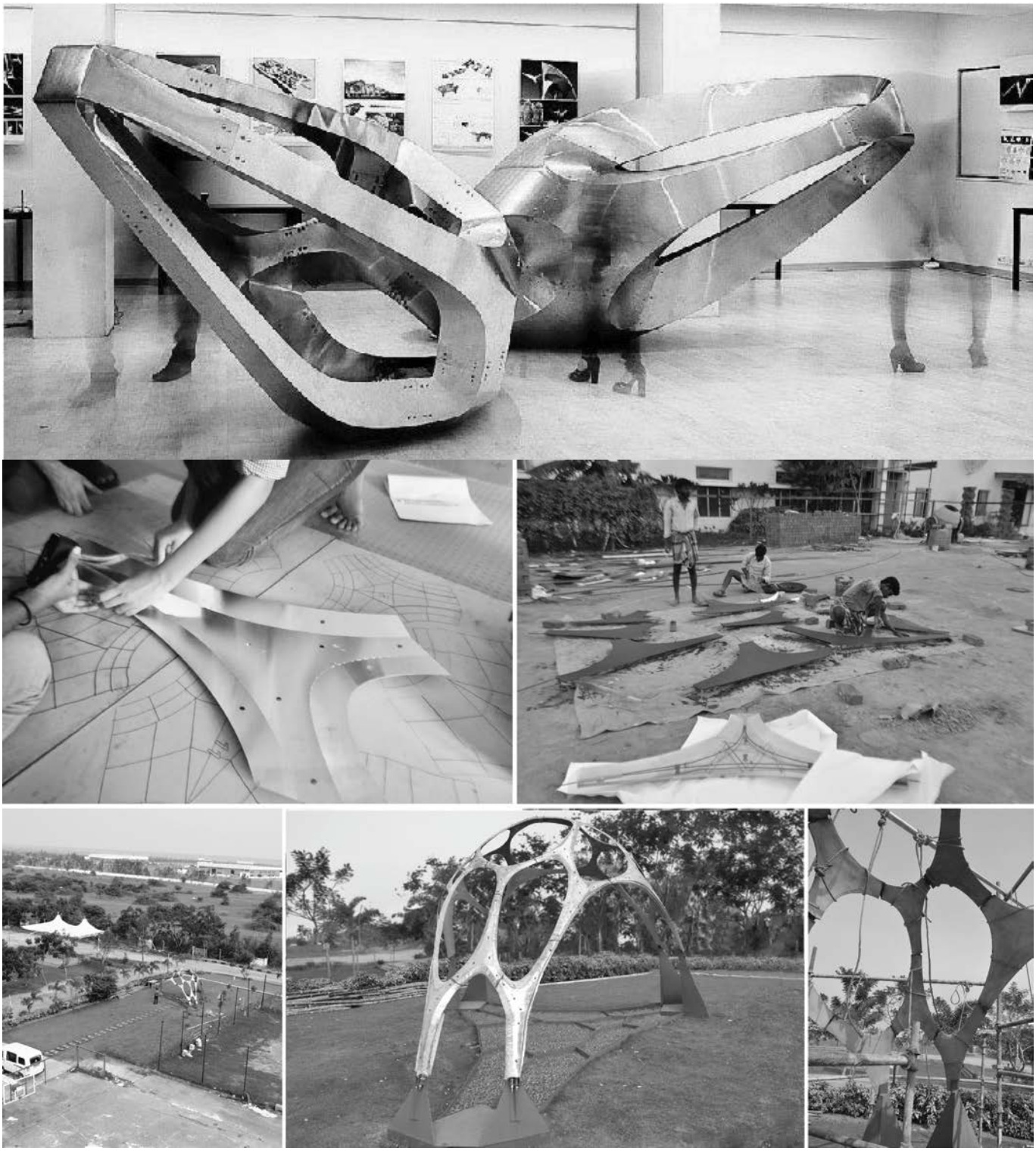
The simulation employs a mass-spring paradigm and thus inherits associated limitations. The simulation assumes thin surface geometries and cannot model the thickness of material. This requires that manufacturing details such as positioning of boltholes, calculation of spring-back etc, are empirically approximated. Another associated limitation of the method that employs virtual or non-physically based forces, is the need to experimentally establish various parameters such spring constrants, weights between forces etc. This becomes amplified when the simulation deals with forces that are in differing units such the developability force being related to angles and the planarity force being derived from distance measurements.

CONCLUSION

The various caveats and limitations of the proposed method are noted in the previous section. As a proof-of concept, the proposed method was successfully employed to design and fabricate a self-supporting structure composed on folded panels, in relatively short span of time – design to completion in 3 months. Further the proposed method was found to be designer-friendly in that it utilisizes popular mesh modeling procedures, thereby easing the assimilation into established contemporary design workflows. The method also allows for multiple possibilities of feedback and iteration within the various steps of the process, thus allowing for multiple collaborative inputs to be assimilated during the process of design. In short, the method allows for integration into a general interactive and iterative *form-finding* framework

that can be employed for *finding* minimal surfaces – both Gaussian-curvature minimal and mean-curvature minimal - along with their planar developments, allowing for their manufacture from sheet material – paper / metal and stretched fabric.

Image 2: Images of built applications of our method: sculpture and cast-concrete structure.



REFERENCES

Aleksandrov, A. and Zalgaller, V. (1967). *Intrinsic geometry of surfaces*. Providence: American Mathematical Society, p.327.

Barnes, M. (1999). Form Finding and Analysis of Tension Structures by Dynamic Relaxation. *International Journal of Space Structures*, 14(2), pp.89-104.

Bhooshan, S. and El sayed, M. (2015). Sub-division surfaces in Architectural form-finding and fabric-forming. In: *2nd international conference on flexible formwork, ICFF 2012*. Bath: The International Society of Fabric Formwork.

Bhooshan, S. and El Sayed, M. (2011). Use of Sub-division Surfaces Architectural Form-Finding and Procedural Modelling. In: *simulation in Architecture and urbanism, simAUD 2011*. Boston.

Conway, J., Burgiel, H. and Goodman-Strauss, C. (2008). *The symmetries of things*. Wellesley, Mass.: A.K. Peters.

Demaine, E., Demaine, M., Koschitz, D. and Tachi, T. (2011). Curved Crease Folding: a Review on Art, Design and Mathematics. In: *Proceedings of the IABSE-IASS Symposium: Taller, Longer, Lighter (IABSE-IASS 2011),*. London.

Eric, D. (n.d.). *Curved-Crease Sculpture by Erik and Martin Demaine*. [online] Erikdemaine.org. Available at: http://erikdemaine.org/curved/ [Accessed 23 Jun. 2013].

Desbrun, M., Meyer, M. and Alliez, P. (2002). Intrinsic Parameterizations of Surface Meshes. *Computer Graphics Forum*, 21(3), pp.209-218.

Hormann, K. and Greiner, G. (2000). MIPS: An Efficient Global Parametrization Method. In: *Curve and Surface Design*. Vanderbilt University Press, pp.153-162.

Huffman, D. (n.d.). *Geometric Paper Folding: Dr. David Huffman*. [online] Graficaobscura.com. Available at: http://www.graficaobscura.com/huffman/ [Accessed 8 Aug. 2013].

Kilian, M., Flöry, S., Chen, Z., Mitra, N., Sheffer, A. and Pottmann, H. (2008). Curved folding. *ACM Trans. Graph.*, 27(3), p.1.

Liu, L., Zhang, L., Xu, Y., Gotsman, C. and Gortler, S. (2008). A Local/Global Approach to Mesh Parameterization. *Computer Graphics Forum*, 27(5), pp.1495-1504.

McCartney, J., Hinds, B. and Seow, B. (1999). The flattening of triangulated surfaces incorporating darts and gussets. *Computer-Aided Design*, 31(4), pp.249-260.

Mitani, J. and Igarashi, T. (2011). Interactive Design of Planar Curved Folding by Reflection. In: *The 19th Pacific Conference on Computer Graphics and Applications (Pacific Graphics 2011),*. Kaohsiung, Taiwan.

Pinkall, U. and Polthier, K. (1993). Computing Discrete Minimal Surfaces and Their Conjugates. *Experimental Mathematics*, 2(1), pp.15-36.

Poranne, R., Ovreiu, E. and Gotsman, C. (2013). Interactive Planarization and Optimization of 3D Meshes. *Computer Graphics Forum*, 32(1), pp.152-163.

Shepherd, P. and Richens, P. (2009). Subdivision Surface Modelling for Architecture. In: *Proceedings International Association for Shell and Spatial Structures Symposium*. Valencia.

Sheffer, A., Praun, E. and Rose, K. (2006). Mesh Parameterization Methods and Their Applications. *FNT in Computer Graphics and Vision*, 2(2), pp.105-171.

Solomon, J., Vouga, E., Wardetzky, M. and Grinspun, E. (2012). Flexible Developable Surfaces. *Computer Graphics Forum*, 31(5), pp.1567-1576.

Stam, J. (1998). Exact Evaluation of Catmull-Clark Subdivision Surfaces at Arbitrary Parameter Values. In: *Proceedings SIGGRAPH 1998*. pp.395-404.

Sweeney, R. (2015). *Richard Sweeney*. [online] Richardsweeney.co.uk. Available at: http://www.richardsweeney.co.uk/ [Accessed 16 Jan. 2013].

Tachi, T. (2009). Simulation of rigid origami. In: *Origami 4*. Massachusetts: A K Peters Ltd, pp.175-187.

Wang, C., Smith, S. and Yuen, M. (2002). Surface flattening based on energy model. *Computer-Aided Design*, 34(11), pp.823-833.

Zhong, Y. and Xu, B. (2006). A physically based method for triangulated surface flattening. *Computer-Aided Design*, 38(10), pp.1062-1073.